

IMPLEMENTASI PROSES UJI SISTEM INFORMASI ADMISI PASCASARJANA UNIVERSITAS SAM RATULANGI

Eko Pandara¹⁾, Stanley Karouw²⁾, Meicsy Najooan³⁾

^{1,2,3} Program Studi Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi

Jl. Kampus UNSRAT Bahu, Manado, 95115

Telp : (0431) 852959, Fax : (0431) 823705

E-mail : ekopandara@ymail.com¹⁾, stanley.karouw@unsrat.ac.id²⁾, meicsynajooan@gmail.com³⁾

Abstrak

Kualitas Perangkat Lunak menjadi hal yang sangat penting di era perkembangan teknologi informasi yang pesat saat ini. Untuk mengukur apakah suatu perangkat lunak berkualitas maka harus dilakukan pengujian. Pengujian (Testing) adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan untuk menemukan defects/ errors/ bugs dan mengevaluasi fitur-fitur dari entitas software. Demikian juga dalam pengembangan Sistem Informasi Admisi Pasca Sarjana UNSRAT perlu dilaksanakan pengujian untuk menemukan bug sehingga kemudian dapat dilaporkan kepada tim pengembang. Penelitian ini bertujuan untuk menemukan dan mendaftarkan bug yang terjadi pada Sistem Informasi Admisi Pasca Sarjana UNSRAT. Metode yang digunakan adalah metode Verifikasi dan Validasi. Yang menjadi inti dari metode ini adalah pengujian. Pengujian dilaksanakan dengan menggunakan teknik-teknik pengujian seperti Required Based Testing, White Box Testing, Black Box Testing, dan Basic Path Testing. Proses Pengujian menggunakan Requirement Based Testing Framework yang digunakan oleh RIM bekerja sama dengan University of Waterloo. Hasil penelitian diperoleh hasil berupa daftar error yang terjadi pada perangkat lunak Sistem Informasi Admisi Pasca Sarjana UNSRAT, dan dalam proses pengujian dihasilkan dokumentasi test plan document, test suite document, test report document. Error yang terjadi adalah kesalahan fungsi beberapa fitur yang tidak berjalan baik, kesalahan pada antarmuka yang mengganggu estetika interaksi manusia dan komputer.

Kata Kunci: Perangkat Lunak, Pengujian, Verifikasi & Validasi

1. PENDAHULUAN

Pressman[1] mendefinisikan Kualitas Perangkat Lunak sebagai: “Suatu proses perangkat lunak yang efektif diterapkan dalam arti kata proses perangkat lunak yang menyediakan nilai yang dapat diukur untuk mereka yang menghasilkan dan untuk mereka yang menghasilkannya”. Sedangkan menurut pandangan De Marco, sesuai yang ditulis Pressman[1] menyatakan: “Kualitas produk sesungguhnya merupakan fungsi dari seberapa banyak ia mengubah dunia menjadi semakin baik”.

Pengujian perangkat lunak merupakan tahapan penting dalam proses pengembangan perangkat lunak. Pengujian menjamin kualitas perangkat lunak, sebagai sebuah produk. Adapun penulisan paper ini dibuat untuk menjawab permasalahan penelitian sebagai berikut: (1). Bagaimana merencanakan pengujian perangkat lunak Sistem Informasi(SI) Admisi Pasca Sarjana UNSRAT? (2). Bagaimana melakukan pengujian perangkat lunak SI Admisi Pasca Sarjana UNSRAT yang dapat menjabarkan secara terukur kualitas SI yang telah dibangun? (3). Bagaimana mendokumentasikan pengujian yang dilakukan pada SI Admisi Pasca Sarjana UNSRAT?

2. LANDASAN TEORI

Pada bagian ini akan dijelaskan beberapa teori terkait dengan implementasi proses uji SI Admisi Pascasarjana Universitas Sam Ratulangi. Metodologi yang digunakan juga dijelaskan pada bagian ini.

2.1 Pengujian dan Kualitas Perangkat Lunak

Pengertian Kualitas Perangkat Lunak (*Software quality*) menurut IEEE *Standard Glossary of Software Engineering Technology* [2] yang dikatakan sebagai: “The degree to which a system, component, or process meets customer or user needs or expectation.” Pendekatan *engineering* menginginkan bahwa kualitas perangkat lunak ini dapat diukur secara kuantitatif, dalam bentuk angka-angka yang mudah dipahami oleh manusia. Untuk itu perlu ditentukan parameter atau atribut pengukuran.

2.2 Verification and Validation (V&V)

Menurut Donald Firesmith[4] verifikasi dan validasi biasanya dilakukan dengan menggunakan satu atau lebih teknik seperti :

1. Analisis : penggunaan model teknis atau matematika, simulasi, algoritma, atau prinsip-prinsip dan prosedur ilmiah untuk menentukan apakah suatu produk kerja memenuhi persyaratan
2. Demonstrasi : pemeriksaan visual dari pelaksanaan produk bekerja di bawah skenario tertentu untuk menentukan apakah memenuhi persyaratan
3. Pemeriksaan : pemeriksaan visual (termasuk manipulasi fisik atau penggunaan pengukuran mekanis yang sederhana) dari produk kerja non-pelaksana untuk menentukan apakah memenuhi persyaratan
4. Pengujian : stimulasi produk karya yang dapat dijalankan dengan input dan prasyarat dikenal diikuti oleh perbandingan yang sebenarnya dengan respon yang dibutuhkan (output dan *post conditions*) untuk menentukan apakah memenuhi persyaratan

2.3 Prinsip Pengujian

Terdapat 6 kunci prinsip-prinsip testing yaitu:

1. Testing yang komplit tidak mungkin.
2. Testing merupakan pekerjaan yang kreatif dan sulit.
3. Alasan yang penting diadakannya testing adalah untuk mencegah terjadinya *errors*.
4. Testing berbasis pada resiko.
5. Testing harus direncanakan.
6. Testing membutuhkan independensi.

3. PEMBAHASAN

Dalam implementasi pengujian SI ini dilaksanakan berdasarkan prosedur penelitian yang telah dirancang sebelumnya. Prosedur ini terdiri atas beberapa tahap yakni tahap evaluasi data, tahap analisa data, tahap perencanaan pengujian, tahap pembentukan kasus uji, tahap eksekusi pengujian dan tahap verifikasi dan validasi hasil. Berikut uraian tiap tahap yang dilaksanakan pada penelitian ini:

3.1 Tahap Evaluasi Data

Tahapan evaluasi data dilaksanakan dengan cara mengumpulkan data utama dan data pendukung. Tujuannya untuk mendapat gambaran secara menyeluruh tentang apa yang sudah dikerjakan oleh orang lain dan bagaimana mereka mengerjakannya dengan mengambil data yang valid untuk digunakan sebagai bahan penelitian. Data-data tersebut adalah hasil rancangan Sistem Informasi Admisi Pasca Sarjana UNSRAT. Hasil dari tahapan ini adalah berupa data-data yang dibutuhkan dalam pengujian perangkat lunak.

3.2 Tahap Analisa Data

Tahap selanjutnya yang dilakukan adalah tahap analisa data. Berdasarkan data-data yang diperoleh pada tahap evaluasi data, dilakukan analisa terhadap data-data yang diperoleh. Analisa disini berupa analisa kebutuhan perangkat lunak dengan meninjau dokumen SRS SI Admisi Pasca Sarjana UNSRAT. Aktor dalam *use case* ini adalah Admin dan Calon Mahasiswa. Calon Mahasiswa dapat mencetak bukti pendaftaran setelah melakukan pendaftaran, apabila terjadi error dalam pencetakan maka Calon Mahasiswa dapat menghubungi Admin untuk mencetak bukti pendaftaran. Berdasarkan *use case* yang ada, fungsi-fungsi yang bisa dilakukan oleh Aktor adalah :

- a) Login sebagai Admin
- b) Login sebagai User
- c) Manage Data
- d) Entry Data
- e) Print Data

Berdasarkan analisa proses bisnis dan kebutuhan sistem diatas diperoleh hasil potensi kesalahan pada perangkat lunak yang bisa terjadi diantaranya sebagai berikut:

(1) Potensi Kesalahan Fungsional adalah kesalahan dalam mengimplementasikan kebutuhan atau spesifikasi desain. Hasil analisa potensi kesalahan fungsional ini menghasilkan target pengujian dan potensi kesalahan yang bisa terjadi. Dengan Pengujian fungsional maka kesalahan yang terjadi pada target yang berpotensi dapat diketahui. (2) Potensi Kesalahan Unit, Kesalahan ini terjadi pada area struktur perangkat lunak, seperti alur program, waktu eksekusi program, dan *source code* program. Area ini dapat diuji dengan Pengujian Unit. (3) Potensi Kesalahan Basis Data, Kesalahan ini berpotensi terjadi pada tipe data dan query pada basis data yang dibuat. Potensi kesalahan pada area ini dapat diuji pengujian basis data. (4) Potensi Kesalahan Antarmuka, Kesalahan ini berpotensi terjadi pada area antarmuka (*interface*). Potensi kesalahan pada area ini dapat diuji dengan pengujian antarmuka.

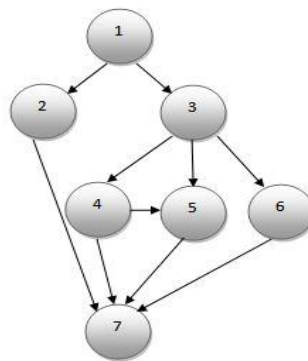
3.3 Tahap Perencanaan Pengujian (Test Plan)

Tahap selanjutnya yang dilakukan adalah ada tahap perencanaan. Tahap ini merupakan tahap awal pada framework pengujian perangkat lunak oleh RIM seperti pada gambar 2. Pada tahap ini dibuat perencanaan pengujian yang meliputi Tujuan Pengujian, Ruang Lingkup Pengujian, Strategi Pengujian, Perangkat Lunak Pengujian (*tools*), Arsitektur Pengujian. Tahap ini menghasilkan dokumentasi berupa dokumen Rencana Pengujian (*Test Plan Document*).

3.4 Tahap Pembentukan Kasus Uji (Test Case Generated)

Tahap selanjutnya setelah perencanaan pengujian adalah tahap pembentukan kasus uji. Pada tahap ini dilakukan pembentukan *test case* pada masing-masing tipe pengujian yakni pembuatan *test case* pengujian fungsional, pengujian database dan antarmuka menggunakan teknik RBT dan *Black Box Testing*, pembuatan *test case* pengujian unit menggunakan teknik *White Box Testing* dan *Basic Path Testing*, pembuatan. Hasil dari tahap ini adalah berupa dokumentasi test suite (*Test Suite Document*).

Pada tahap ini dilakukan pembentukan *test case* dengan teknik membuat *Flow Graph* terlebih dahulu. Berdasarkan *use case* terdapat dua aktor yang berperan dalam sistem yaitu, Admin dan Calon_mahasiswa. *Graph* dibuat sesuai dengan aktifitas yang dilakukan oleh masing-masing aktor. *Graph* dibuat dengan asumsi seluruh proses hanya berjalan searah, tidak ada proses yang looping dan minimal setiap node dilewati sekali. Tujuan pembuatan *graph* adalah untuk menentukan minimum path coverage untuk mencapai full line coverage. Dengan demikian, *test case* yang dibuat merupakan kumpulan *test case* minimum yang dapat memenuhi seluruh modul fungsi yang ada. Bentuk *graph* dari fungsi aktor Admin dapat dilihat pada gambar 1:



Gambar 1 Graph fungsi Admin

Keterangan tiap node pada graph:

1. Login
2. Tambah data user
3. Lihat data Calon_Mahasiswa
4. Edit data Calon_Mahasiswa
5. Cetak data Calon_Mahasiswa
6. Hapus data Calon_Mahasiswa
7. Logout

Berdasarkan *flow graph* tersebut terdapat 5 kemungkinan *path* yang ada dengan jumlah edge sebanyak 10. Kemungkinan seluruh *path* adalah:

Tabel 3.3 Jalur Path fungsi Admin

Jalur	Path
1	1-2-7
2	1-3-4-7
3	1-3-4-5-7
4	1-3-5-7
5	1-3-6-7

Dengan menggunakan rumus *cyclomatic complexity metric* dapat dihitung nilai kompleksitas dari blok program yang akan diuji. Berikut perhitungannya:

$$E = 10$$

$$N = 7$$

$$P = 1$$

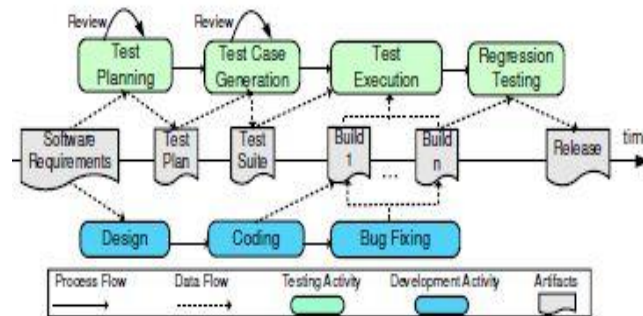
$$CC = E - N + P$$

$$CC = 10 - 7 + 1 = 4$$

Nilai kompleksitas dari blok program ini adalah 4.

3.5 Tahap Pengujian

Eksperimen dan Pengujian menggunakan teknik Pre-experimental dengan teknik *One-Shot Case Study*. Peneliti memilih Pengujian hanya dilakukan 1 kali, pada tahap awal fase konstruksi oleh tim pengembang. Proses Pengujian diadaptasi dari Model Pengujian yang dilakukan oleh Perusahaan Telekomunikasi RIM. Proses pengujian tersebut dapat dilihat pada gambar 2 dibawah ini:



Gambar 2 High Level Testing Process Model at RIM

Dalam pengujian ini terdapat 4 tipe pengujian yang dilaksanakan yakni:

(1) Pengujian Fungsional

Pada tahap ini dilakukan pembentukan kasus pengujian. Hasil analisa terhadap *Use Case* yang dibuat oleh tim pengembang SI terdapat 5 fungsi utama dari Sistem, yaitu *login admin*, *manage data*, *print data*, *login user*, dan *entry data*.

(2) Pengujian Unit

Pengujian unit ini dilaksanakan dengan menggunakan bahasa pemrograman Java yang terintegrasi dengan JUnit dan Selenium *Remote Control* (Java/JUnit/RC). Pengujian ini dimulai dengan melakukan Generate terhadap hasil pengujian Fungsional di Selenium. Dengan bantuan selenium kasus pengujian yang sebelumnya berupa alur kerja program diubah menjadi bahasa pemrograman java untuk pengujian menggunakan *JUnit Testing* pada eclipse.

(3) Pengujian Basis Data

Implementasi pengujian basis data dilakukan dengan prosedur pengujian database seperti pada tabel dibawah ini:

Tabel 1 Pengujian Basis Data

Objective:	Memastikan data tabel master dan transaksi serta relationalnya berjalan dnegan baik.
Technique:	<p>1. Mengecek relational tabel master dengan transaksi Input : memasukkan data di tabel master dan tabel transaksi Proses : menghapus <i>record</i> data master Output : data tidak boleh terhapus,jika terhapus mengandung arti terdapat relasi yang salah</p> <p>2. Melakukan query dengan banyak tabel Input : jika di mungkinkan,menggunakan perintah "<i>select</i>" dengan mengambil dari semua data master dan transaksi yang berelasi Proses : melakukan <i>query</i> Output : data harus tampil</p> <p>3. Mengecek tipe data dari masing-masing <i>field</i> Input : memberikan inputan berupa huruf pada tipe data "<i>date/time</i>" Proses : mencoba secara langsung pada <i>interface</i> database seperti "<i>phpmyadmin</i>" didalam RDBMS mysql Output : data tidak boleh tersimpan</p> <p>4. Mengecek Atribut dari masing-masing tabel Input : memasukkan data yang sama pada atribut <i>primary key</i> atau <i>unique</i> dalam suatu <i>record</i> baru Proses : melakukan <i>query</i> input data "<i>insert</i>" Output : tidak boleh ada yang sama pada <i>field</i> beratribut <i>primary key</i></p>
Completion Criteria:	Setiap perintah query dapat berjalan tanpa terjadi kesalahan
Special Considerations:	<i>Query</i> untuk mengambil data pada semua tabel hanya sebagai percobaan dan jika itu di mungkinkan sesuai dengan kondisi relational yang ada.

(4) Pengujian AntarMuka (*User Interface Testing*)

Pengujian ini dilakukan dengan mengidentifikasi kesalahan yang terjadi pada antarmuka perangkat lunak Sistem Informasi Admisi Pasca Sarjana UNSRAT.


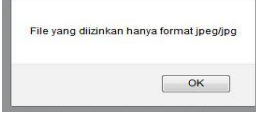

Tabel 2 Pengujian AntarMuka

<i>Test Objective:</i>	Menguji komponen menu, ukuran, posisi, titik fokus tampilan, kemudahan penggunaan serta aksesibilitasnya.
<i>Technique:</i>	<ol style="list-style-type: none"> Mencoba setiap form yang ada baik dari objek maupun tampilannya. Input : menginputkan data berbagai tipe karakter Proses : input data Output : adanya kesesuaian dari data yang di inputkan dengan data yang ada di database. Mencoba navigasi dari semua komponen di dalam website Input : membuka semua menu secara berurutan Proses : klik, drag and drop Output : harus memberikan kemudahan bagi pengguna serta memiliki kecepatan yang baik saat membuka website. Memastikan ukuran dari website dapat berjalan secara baik pada banyak browser dan sistem operasi. Input : membuka website dengan minimal 3 browser berbeda Proses : membuka halaman website Output : membandingkan antara browser yang satu dengan yang lain.
<i>Completion Criteria:</i>	Gambaran tampilan dapat dengan mudah di jalankan berdasarkan pembandingan (<i>banchmark</i>) yang tepat.
<i>Special Considerations:</i>	-

3.6 Tahap Evaluasi dan Validasi

Pengujian yang dilakukan merupakan pengujian pada tahapan awal setelah pembangunan sistem. Kesalahan yang terjadi dapat berupa *software error* yang dibuat oleh programmer. Kesalahan ini bisa menjadi kesalahan *source code* dalam satu atau lebih dari baris kode, atau kesalahan logis dalam melaksanakan satu atau lebih dari kebutuhan klien. Berikut adalah daftar *Bug* yang ditemukan selama pengujian SI Admisi Pasca Sarjana UNSRAT:

Tabel 3 Daftar *Bug* SI Admisi Pasca Sarjana UNSRAT

NO	<i>Bug</i>	Tipe Pengujian	Keterangan
1	Tidak bisa input transkrip format jpg	Fungsional	 <p>Muncul pesan <i>error</i> seperti diatas, meskipun format yang dimasukan adalah format jpg</p>
2	Tidak bisa input ijazah format jpg	Fungsional	 <p>Muncul pesan <i>error</i> seperti diatas, meskipun format yang dimasukan adalah format jpg</p>
3	id=jenang not found	Unit	id=jenang tidak terdefinisi
4	Tampilan yang berubah Pada Antarmuka	Antarmuka	<p>Pada browser eksplorer tampilan berubah</p> 

Hasil pengujian di atas dapat dijadikan sebagai pedoman untuk pengujian selanjutnya. Dikarenakan pengujian pada yang dilakukan adalah pengujian awal pada proses pengembangan perangkat lunak.\

4. SIMPULAN DAN SARAN

Berdasarkan hasil laporan penelitian ini, penulis dapat menarik kesimpulan yaitu sebagai berikut:

- 1) Model Proses Pengujian Verifikasi & Validasi dapat diterapkan pada aplikasi web dengan menggunakan perangkat lunak pengujian Selenium.
- 2) Jenis *bug* yang ditemukan adalah kesalahan fungsi beberapa fitur yang tidak berjalan sesuai harapan, kesalahan pada antarmuka yang mengganggu estetika interaksi manusia dan komputer. *Bug* yang ditemukan adalah tidak bisa input transkrip format jpg, tidak bisa input ijazah format jpg, id=jenjang not found, tampilan yang berubah pada antarmuka pada browser yang berbeda.
- 3) Teknik dan *tools* pengujian selalu digunakan secara bersama untuk menghasilkan reliabilitas hasil uji.

Saran untuk penelitian kedepannya:

- 1) Untuk pengujian kedepannya dapat ditambahkan *acceptance testing* pada sistem ini dimana pengujian ini dilakukan oleh user Beta
- 2) Setiap *bug* dan *error* yang ditemukan perlu diperbaiki
- 3) Uji pada Live-Apps-Error

5. DAFTAR RUJUKAN

- [1] Pressman, Software Engineering, A Practitioner's Approach, 6th ed, McGrawHill, Singapura, 2005.
 - [2] IEEE Standard Glossary of Software Engineering Technology, IEEE Std 610.12-1990, Institute of Electrical and Electronics Engineers, New York, 1990.
 - [3] Sommerville, Software Engineering, 8th ed, Pearson Education Limited, 2007
 - [4] Donald G. Firesmith, "Quality Assessment of System Architectures and their Requirements (QUASAR), Version 3.1," 30 minute presentation at the 12th Annual NDIA Systems Engineering Conference in San Diego California on 29 October 2009, p. 64.
-